

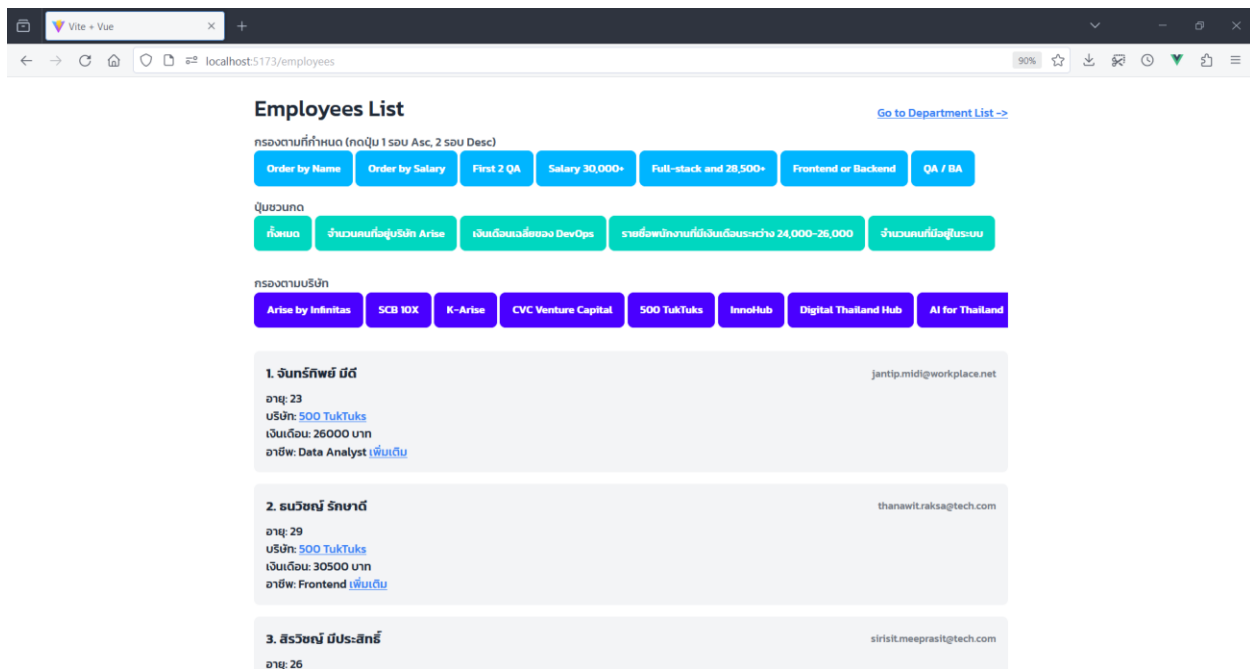
INT305 – Class Assignment

SID: [64130500062](#)

Name: Pakapol Taubol (ภคพล ทาอุบล)

Application name: Employees List Query – App

Description: แอปนี้ทำเพื่อวิชา INT305 โดยผมได้ศึกษาการทำ Query ต่างๆ และความต้องการของอาจารย์เอาไว้ และได้ออกมาเป็นแอปนี้ครับ โดยตัวแอปจะมีปุ่มให้กดเพื่อเป็นการ Query โดยรายชื่อจะเปลี่ยนไปตาม Query



ข้อแนะนำ: ถ้าหากอาจารย์เกิดมีข้อผิดพลาดในการโหลดข้อมูล โปรดให้อาจารย์เปลี่ยน Firebase config เป็นของตัวเองและทดสอบได้เลย แต่ถ้าหากอาจารย์ไม่สะดวกสามารถทักหาผมที่ MS Team pakapol.taub@kmutt.ac.th ได้เลยครับ

ภาพรวมของแอป

Vite + Vue

localhost:5173/employees

80%

Employees List

[Go to Department List ->](#)

กรองงานที่ทำงาน (กดปุ่ม 1 สอง Asc, 2 สอง Desc)

Order by Name

Order by Salary

First 2 QA

Salary 30,000+

Full-stack and 20,500+

Frontend or Backend

QA / BA

ปุ่มรวมข้อมูล

ทั้งหมด

จำนวนคนที่อยู่ในบริษัท Arise

เงินเดือนเฉลี่ยของ DevOps

รายชื่อพนักงานที่มีเงินเดือนระหว่าง 24,000-26,000

จำนวนคนที่อยู่ในระบบ

กรองตามบริษัท

Arise by Infinitas

SCB IOX

K-Arise

CVC Venture Capital

500 TukTuk

Innohub

Digital Thailand Hub

AI for Thailand

1. ทศพร มาภูษา

pakapol.taubgkmutt.ac.th

อายุ: 21
บริษัท: [Arise by Infinitas](#)
เงินเดือน: 25000 บาท
อาชีพ: Full-stack [เพิ่มเติม](#)

2. สุภาวดี แซงัด

supawadee.sand@gmail.com

อายุ: 25
บริษัท: [K-Arise](#)
เงินเดือน: 28000 บาท
อาชีพ: Frontend [เพิ่มเติม](#)

3. วิชยกรม ประดิษฐ์

wichayakorn.prag-company.com

อายุ: 30
บริษัท: [InnoHub](#)
เงินเดือน: 35000 บาท
อาชีพ: Backend [เพิ่มเติม](#)

Vite + Vue

localhost:5173/departments

80%

Department List

[Go to Employees List ->](#)

กรองงานสเกลที่คุ้นเคย

HTML, CSS, JavaScript

Excel

SQL

Python

1. BA

สเกลที่คุ้นเคย: [Excel, PowerPoint, SQL](#)

2. Backend

สเกลที่คุ้นเคย: [Java, SQL](#)

3. Data Analyst

สเกลที่คุ้นเคย: [Excel, SQL, Power BI, Tableau](#)

4. Data Engineer

สเกลที่คุ้นเคย: [SQL, Hadoop, Spark, Machine Learning](#)

5. Data Scientist

สเกลที่คุ้นเคย: [Python, SQL, Machine Learning, Deep Learning](#)

6. DevOps

สเกลที่คุ้นเคย: [Docker, Kubernetes, Git, CI/CD](#)

7. Frontend

สเกลที่คุ้นเคย: [HTML, CSS, JavaScript, React.js](#)

8. Full-stack

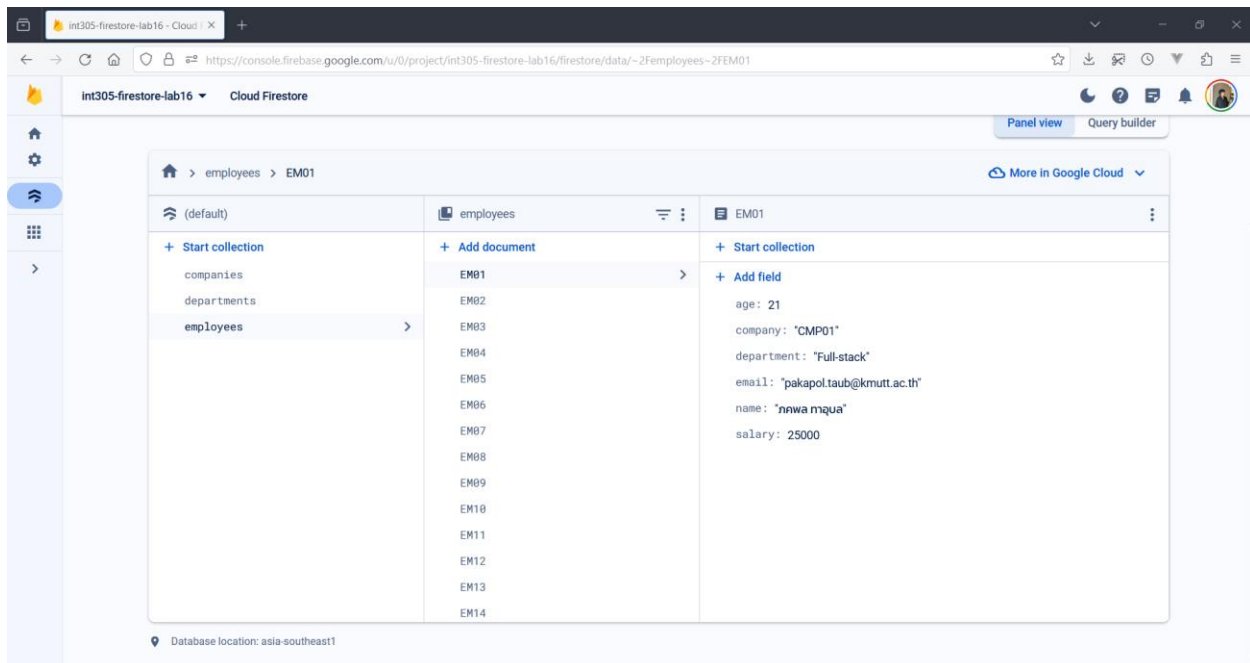
ข้อมูลในระบบ

ข้อมูลในระบบของพม นั้นจะมีทั้งหมด 3 collection

1. Employees – พนักงาน (ตั้งค่า Doc ID เองเช่น EM01)
2. Companies – บริษัท (ตั้งค่า Doc ID เองเช่น CMP01)
3. Departments – สาขา/อาชีพ (ตั้งค่า Doc ID เป็นตามชื่ออาชีพนั้นๆ)

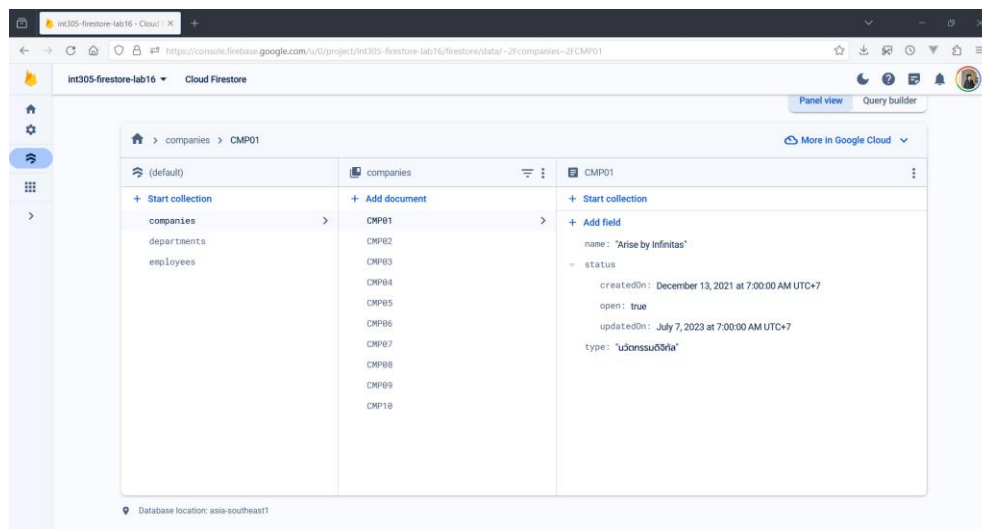
โดยข้อมูล Employees

1. Doc ID (EM01-EMXX)
2. name(string) ชื่อของพนักงาน
3. age(number) อายุของพนักงาน
4. company(string) รหัสบริษัทของพนักงาน
5. department(string) สาขา/อาชีพ ของพนักงาน
6. email(string) อีเมลของพนักงาน
7. salary(number) เงินเดือนของพนักงาน



ข้อมูล companies

1. Doc ID (CMP01-CMPXX)
2. name(string) ชื่อบริษัท
3. type(string) ประเภทของบริษัท
4. status(Map/Object):
 - open(Boolean) สถานะของบริษัท
 - createdOn(Timestamp) วันที่ก่อตั้งของบริษัท
 - updatedOn(Timestamp) วันที่อัปเดตล่าสุดของบริษัท



ข้อมูล departments

1. Doc ID (ชื่อเดียวกันกับชื่ออาชีพ)
2. name(string) ชื่อของอาชีพ
3. skill(array) สกิลที่ต้องมีของอาชีพนี้

🏠 > departments > BA		
🏠 (default)	📁 departments	📁 BA
+ Start collection	+ Add document	+ Start collection
companies	BA >	+ Add field
⋮ departments >	Backend	name: "BA"
employees	Data Analyst	▼ skill
	Data Engineer	0 "Excel"
	Data Scientist	1 "PowerPoint"
	DevOps	2 "SQL"
	Frontend	
	Full-stack	
	HR	
	Marketing Manager	
	Product Manager	
	QA	
	Sales Manager	

Design Pattern

Design Pattern: Facade

Facade เป็นหนึ่งใน Design Pattern ในการเขียนโปรแกรมที่มุ่งเน้นไปที่การจัดการและปรับปรุงความซับซ้อนของระบบ โดยการให้ Interface ที่เป็นมีที่นั่นทำหน้าที่เป็นตัวแทนเพื่อลดความซับซ้อนของการเรียกใช้งานส่วนอื่น ๆ ในระบบหรือ Subsystem ต่าง ๆ ซึ่งช่วยให้โปรแกรมง่ายต่อการเข้าใจและบำรุงรักษาเมื่อมีการเปลี่ยนแปลงใน Subsystem หรือ โครงสร้างภายในของระบบโดยไม่ต้องทำการเปลี่ยนแปลง Interface ภายนอก

ดีไซน์ของผมจะมีลักษณะแบบ Facade โดยดีไซน์เน้นไปที่การลดความซับซ้อนของระบบ และระบบสามารถที่จะเชื่อมต่อหรือเข้าถึงได้ง่าย

ตัวอย่าง ใน collection Employees จะมีข้อมูล company และ department ที่สามารถระบุได้ว่า พนักงานคนที่ทำงานที่บริษัทไหน และอาชีพของเขาเป็นอย่างไร ทำให้สามารถเข้าถึงตารางอื่นๆ ได้ง่าย

ประโยชน์:

- ลดความซับซ้อนในการเข้าถึงข้อมูล
- ช่วยลดการเขียนโค้ดซ้ำกัน
- เพิ่มความยืดหยุ่นในการปรับเปลี่ยนโค้ด

ERD

employees
Doc id (string)
name (string)
age (number)
email (string)
salary (number)
company (string)
department (string)








companies
Doc id (string)
name (string)
type (string)
status (map/object)
open (boolean)
createdOn (timestamp)
updatedOn (timestamp)

departments
Doc id (string)
name (string)
skill (array)



การ Import ข้อมูล (กรณีจะใช้ Key ของตนเอง)

เข้า Folder ที่ชื่อ import

 .gitignore	23-Nov-23 00:00	Git Ignore Source ...	1 KB
 import-companies	07-Dec-23 19:52	JavaScript Source ...	4 KB
 import-department	07-Dec-23 21:37	JavaScript Source ...	2 KB
 import-employees	07-Dec-23 21:46	JavaScript Source ...	9 KB
 package	22-Nov-23 22:38	JSON Source File	1 KB
 package-lock	07-Dec-23 19:05	JSON Source File	155 KB
 serviceAccountKey	09-Dec-23 22:16	JSON Source File	3 KB

npm install

ในไฟล์ **serviceAccountKey.js** ให้แก้เป็น key ของตัวเอง

node import-companies

node import-department

node import-employees

ตัวอย่าง Query

- พนักงานที่ทำงาน QA สัก 2 คน (limit)

First 2 QA

```
case 2:  
  qry = query(colEmployee, where('department', '=', 'QA'), limit(2));  
  break;
```

1. สุวัฒน์ วรเชษฐ์

suwat.vor@service.org

อายุ: 29

บริษัท: [Digital Thailand Hub](#)

เงินเดือน: 29000 บาท

อาชีพ: QA [เพิ่มเติม](#)

2. สมชาย ประดิษฐ์

somchai.pra@service.org

อายุ: 28

บริษัท: [AI for Thailand](#)

เงินเดือน: 29500 บาท

อาชีพ: QA [เพิ่มเติม](#)

2. พนักงานที่มีเงินเดือน 30,000 บาทขึ้นไปและสามารถเรียงจากมาก-น้อย หรือน้อย-มากได้ (กดปุ่ม 1 รอบ Asc, 2 รอบ Desc) (order By)

Salary 30,000+

case 3:

```
qry = query(colEmployee, where('salary', '>=', 30000), orderBy('salary', isAsc ? 'asc' : 'desc'));  
break;
```

1. ณัฐกฤต สุขสุข

nuttakritsuk@gmail.com

อายุ: 28

บริษัท: [Arise by Infinitas](#)

เงินเดือน: 30000 บาท

อาชีพ: Full-stack [เพิ่มเติม](#)

2. วรวัฑ สุขสวัสดิ์

woravut.sukswasdi@startup.io

อายุ: 28

บริษัท: [SCB 10X](#)

เงินเดือน: 30000 บาท

อาชีพ: Product Manager [เพิ่มเติม](#)

3. ธนวิทย์ รักชาติ

thanawit.raksa@tech.com

อายุ: 29

บริษัท: [500 TukTuks](#)

เงินเดือน: 30500 บาท

อาชีพ: Frontend [เพิ่มเติม](#)

4. วิชิต พรหมทอง

wichit.promthong@corporation.org

อายุ: 29

บริษัท: [Digital Thailand Hub](#)

เงินเดือน: 31000 บาท

อาชีพ: BA [เพิ่มเติม](#)

3. พนักงาน Full-stack ที่มีเงินเดือน 28,500 ขึ้นไป (and , order By)

Full-stack and 28,500+

```
case 4:
  qry = query(colEmployee, and(where('department', '=', 'Full-stack'), where('salary', '>=', 28500)), orderBy('salary', isAsc ? 'asc' : 'desc'));
  break;
```

1. ณภัทร ทองสุข

naphat.thongsuk@company.net

อายุ: 26

บริษัท: [Microsoft Thailand](#)

เงินเดือน: 28500 บาท

อาชีพ: Full-stack [เพิ่มเติม](#)

2. ณัฐกรฤต สุขสุข

nuttakrit.suk@gmail.com

อายุ: 28

บริษัท: [Arise by Infinitas](#)

เงินเดือน: 30000 บาท

อาชีพ: Full-stack [เพิ่มเติม](#)

4. พนักงานที่ทำงาน Frontend หรือ Backend (or)

Frontend or Backend

```
case 5:
  qry = query(colEmployee, or(where('department', '=', 'Frontend'), where('department', '=', 'Backend')));
  break;
```

1. สุภาวดี แสงดี

supawadee.sande@gmail.com

อายุ: 25

บริษัท: [K-Arise](#)

เงินเดือน: 28000 บาท

อาชีพ: Frontend [เพิ่มเติม](#)

2. วิชญากร ประดิษฐ์

wichayakorn.pra@company.com

อายุ: 30

บริษัท: [InnoHub](#)

เงินเดือน: 35000 บาท

อาชีพ: Backend [เพิ่มเติม](#)

3. ปัทรวรรณ พรหมประสิทธิ์

patrawan.pomp@domain.com

อายุ: 27

บริษัท: [SCB 10X](#)

เงินเดือน: 32000 บาท

อาชีพ: Frontend [เพิ่มเติม](#)

5. พนักงานที่ทำงาน QA หรือ BA (in)

QA / BA

```
case 6:  
  qry = query(colEmployee, where('department', 'in', ['QA', 'BA']));  
  break;
```

1. สุวัฒน์ วรเชษฐ

suwat.vor@service.org

อายุ: 29

บริษัท: [Digital Thailand Hub](#)

เงินเดือน: 29000 บาท

อาชีพ: QA [เพิ่มเติม](#)

2. วิชิต พรหมทอง

wichit.promthong@corporation.org

อายุ: 29

บริษัท: [Digital Thailand Hub](#)

เงินเดือน: 31000 บาท

อาชีพ: BA [เพิ่มเติม](#)

3. กฤษฏา อัครวุฒิ

krita.akkarawut@business.co

อายุ: 27

บริษัท: [K-Arise](#)

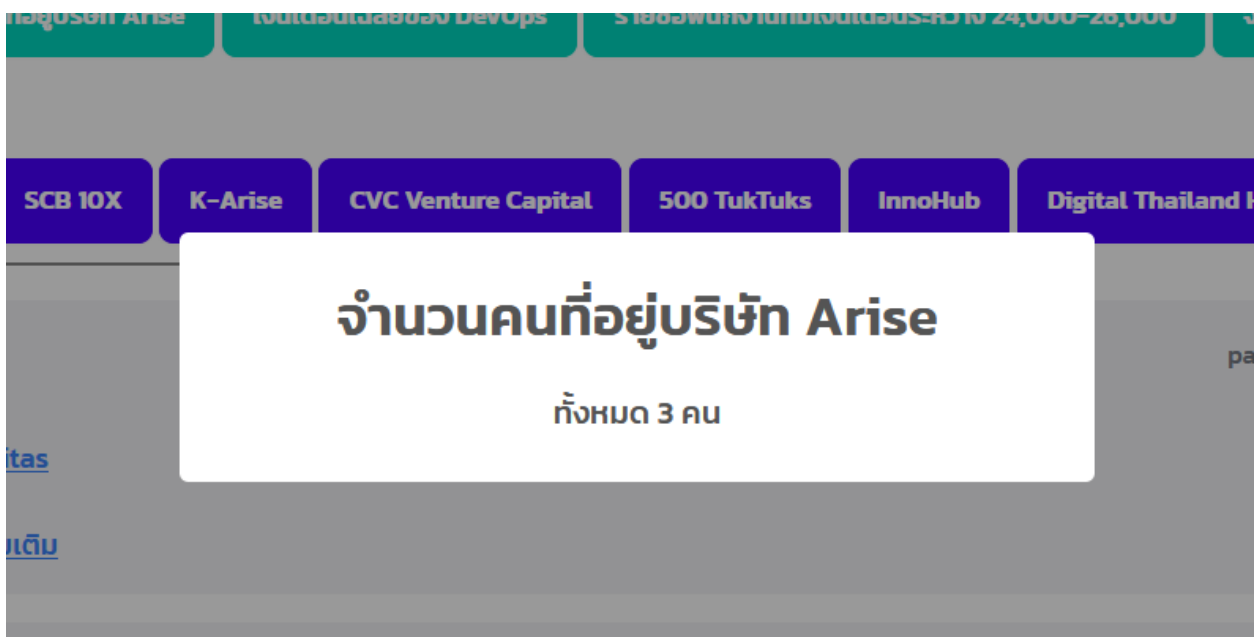
เงินเดือน: 32000 บาท

อาชีพ: BA [เพิ่มเติม](#)

6. พนักงานทั้งหมดที่ทำงานที่ Arise (Aggregate , count)

จำนวนคนที่อยู่บริษัท Arise

```
case 1:
  qry = query(colEmployee, where('company', '==', 'CMP01'));
  snapshotText = await getAggregateFromServer(qry, { total: count('company') });
  text = snapshotText.data().total;
  Swal.fire({
    title: 'จำนวนคนที่อยู่บริษัท Arise',
    text: `ทั้งหมด ${text} คน`,
    showConfirmButton: false
  })
  break;
```



Query ที่นิยม

getData.js:

```
if (mode == 'employees') {
  switch (index) {
    case 0:
      qry = query(colEmployee, orderBy('name', isAsc ? 'asc' : 'desc'));
      break;
    case 1:
      qry = query(colEmployee, orderBy('salary', isAsc ? 'asc' : 'desc'));
      break;
    case 2:
      qry = query(colEmployee, where('department', '=', 'QA'), limit(2));
      break;
    case 3:
      qry = query(colEmployee, where('salary', '>=', 30000), orderBy('salary', isAsc ? 'asc' : 'desc'));
      break;
    case 4:
      qry = query(colEmployee, and(where('department', '=', 'Full-stack'), where('salary', '>=', 28500)), orderBy('salary', isAsc ? 'asc' : 'desc'));
      break;
    case 5:
      qry = query(colEmployee, or(where('department', '=', 'Frontend'), where('department', '=', 'Backend')));
      break;
    case 6:
      qry = query(colEmployee, where('department', 'in', ['QA', 'BA']));
      break;
  }
} else if (mode == 'condition') {
  switch (index) {
    case 0:
      qry = query(colEmployee);
      break;
    case 1:
      qry = query(colEmployee, where('company', '=', 'CMP01'));
      snapshotText = await getAggregateFromServer(qry, { total: count('company') });
      text = snapshotText.data().total;
      Swal.fire({
        title: 'จำนวนคนที่อยู่บริษัท Arise',
        text: `ทั้งหมด ${text} คน`,
        showConfirmButton: false
      });
      break;
    case 2:
      qry = query(colEmployee, where('department', '=', 'DevOps'));
      snapshotText = await getAggregateFromServer(qry, { total: average('salary') });
      text = snapshotText.data().total;
      Swal.fire({
        title: 'เงินเดือนเฉลี่ยของ DevOps',
        text: `ทั้งหมด ${text} บาท`,
        showConfirmButton: false
      });
      break;
    case 3:
      qry = query(colEmployee, and(where('salary', '>=', 24000), where('salary', '<=', 26000)), orderBy('salary', isAsc ? 'asc' : 'desc'));
      break;
    case 4:
      qry = query(colEmployee);
      snapshotText = await getAggregateFromServer(qry, { total: count('company') });
      text = snapshotText.data().total;
      Swal.fire({
        title: `จำนวนทั้งหมด ${text} คน`,
        showConfirmButton: false
      });
  }
}
```

```

else if (mode == 'departments') {
  switch (index) {
    case 0:
      qry = query(colDepartment, where('skill', 'array-contains-any', ['HTML', 'CSS', 'JavaScript']));
      break;
    case 1:
      qry = query(colDepartment, where('skill', 'array-contains', 'Excel'));
      break;
    case 2:
      qry = query(colDepartment, where('skill', 'array-contains', 'SQL'));
      break;
    case 3:
      qry = query(colDepartment, where('skill', 'array-contains', 'Python'));
      break;
    case 4:
      break;
  }
}

```

```

export const getDataByCollectionName = async (col_name) => {
  const col = collection(db, col_name);
  const snapshot = await getDocs(col);
  const data = [];

  for (const doc of snapshot.docs) {
    const docData = doc.data();
    docData.id = doc.id;
    data.push(docData);
  }
  return data;
}

```

```

export const getCompanyNameById = async (cmp_id) => {
  const col = collection(db, "companies");
  const qry = query(col, where(documentId(), '==', cmp_id));
  const snapshot = await getDocs(qry);
  let data = null;

  for (const doc of snapshot.docs) {
    const docData = doc.data();
    docData.id = doc.id;
    data = docData;
  }

  return data.name;
}

```

```
export const queryEmployeesByCompanyID = async (cmp_id) => {  
  const col = collection(db, "employees");  
  const qry = query(col, where('company', '==', cmp_id));  
  const snapshot = await getDocs(qry);  
  const data = [];  
  
  for (const doc of snapshot.docs) {  
    const docData = doc.data();  
    docData.id = doc.id;  
    data.push(docData);  
  }  
  return data;  
}
```